What's going on with

Computational Statistics

Part I: Simulating Statistical Models

1. Uniform (Pseudo) Random Number Generation

The main goal is to produce random variables of a given distribution. Practically it's impossible to generate randomly, so we focus on pseudo random number generators (RNG) of numbers uniformly distributed on the interval $[0,1]$. *Some common pseudo RNGs are:* (1) *linear congruential generator (LCG),* (2) *matrix congruential generator (MCG),* (3) *combined generators,* (4) *"modulo 2" linear generators. How can we check the quality of a RNG? Empirically, using non-parametric goodness-of-fit tests. The idea is always to compare the theoretical cumulative distribution function* $F(\cdot)$ *with the empirical one* $F_n(\cdot)$. *Some methods are the QQ-plot, the Kolmogorov-Smirnov test or the* χ^2*-test.*

2. Random Variable Generation

2.1. Inverse transform methods. *[for "invertible"* $f(\cdot)$*]*

We focused on the uniform distribution but the goal is to construct pseudo RNGs for other distributions than the uniform (starting from the uniform). Many distributions are related to the uniform through simple transformations (or generally they're related to each other). We'll take advantages of these transformations using the inverse transform methods: we generate the random uniform variable U *and through the cumulative distribution function we obtain a value of* $X \sim$ *generic distribution. This is possible both in the discrete and continuous case, but only for one-dimensional cases.*

2.2. Composition method. *[for composed* $f(\cdot)$*]*

We can have a case in which a random variable X *has a mixture cumulative density function* $(F(\cdot))$.
We can write $F(x) = \sum_{i=1}^{n} p_i F_i(x)$ *with* $\{p_i\}_i$ *probability distribution on* $\{F_1, .., F_n\}$. *The idea is to decide from which* $F_i(\cdot)$ *to sample according to the probabilities* p_i *and then sample with one of the previous methods. (e.g. Consider the sum of 3 gaussian:* $\alpha_1 N(\mu_1, \sigma_1^2) + \alpha_2 N(\mu_2, \sigma_2^2) + \alpha_3 N(\mu_3, \sigma_3^2)$. *With probability* α_1 *we'll use* $N(\mu_1, \sigma_1^2)$ *to sample. We generate a random varibale distributed as* $\mathbb{P}(\text{use } N(\mu_i, \sigma_i^2)) = \alpha_i$, *we do a random realization (through previous method), we choose the* $N(\cdot, \cdot)$ *and we sample (through previous methods)).*

2.3. Acceptance-Rejection (A-C) method. *[for un-samplable* $f(\cdot)$*]*

Consider $X \sim f(x)$. *Suppose that we are not able to sample directly from* $f(\cdot)$ *(it may be difficult to invert or we might know* $f(\cdot)$ *up to a constant). We can use an ausiliar distribution* $g(\cdot)$, *easy to sample and such that* $Cg(x) \geq f(x)$, *and we can sample from* $g(\cdot)$. *Once we have the samples* $\{x\}$ *we reject the ones that belong to* $(f(x), Cg(x)]$. *In this way, the accepted samples will be distributed according to the distribution* $f(\cdot)$.

2.4. Transformation of random variables.

Another useful method to sample variables is through the transformations of PDFs, i.e. transformations of random variables. The idea is obtaining a sample from $f(\cdot)$ *starting from* $\tilde{f}(\cdot)$ *of the same location family. The general scheme is* $f(x; \mu, \sigma) = \frac{1}{\sigma} \tilde{f}(\frac{x-\mu}{\sigma})$. *This can be used for example to sample a generic gaussian random variable* $N(\mu, \sigma^2)$*: we sample from* $N(0,1)$ *and then we do an affine transformation. But how can we sample from* $N(0,1)$*? One way is through the Box-Muller transformation, a second way is through the A-C method with* $\mathcal{E}(1)$ *as proposal.*

2.5. Multivariate random variable generation.

In the multivariate case we have to sample $\vec{X}$. *We restrict the study on few situations.*

1. *Independent components. We can generate every* X_i *independently of the others using previous methods.*
2. *Conditional distributions. We want to generate* X_i *knowing the explicit distribution of* $X_i | X_{i-1}, .., X_1$. *We generate a sample of* X_1 *and at every iteration* i *we generate* X_i *based on the conditional distribution.*
3. *Generation by transformation (or copulas). We want to generate samples from* $\vec{X}$, *which has a generic multivariate distribution and dependent components. The idea is to use copulas, i.e. CDFs of dependent uniform random variables. This time, through the copulas, we take into consideration also the dependcy of the variables: we're giving to the generation of* $(U_1, .., U_n)$ *the job to make the components dependent and then we easily pass to the generic requested (multivariate) distribution.*
4. *Multivariate Gaussian distribution. We want to generate* $\vec{X} \sim N(\vec{\mu}, \Sigma)$. *We factorize the covariance matrix* $\Sigma = AA^T$ *and obtain* $\vec{X}$ *as a transformation of* $\vec{Y} \sim N(\vec{0}, I)$ *(i.e.* $Y_i \sim N(0,1)$*):* $\vec{X} = \mu + A\vec{Y}$.
5. *Conditional Multivariate Gaussian distribution. We want to generate samples from* $\vec{X} = (X_1, .., X_n)$ *but only* $(X_{k+1}, .., X_n) := \vec{Z}$ *are observable. The idea is to generate* $(X_1, .., X_k) := \vec{Y}$ *through the conditional distribution* $\vec{Y} | \vec{Z} = \vec{z}$, *which we know is gaussian.*

2.6. Gaussian Process (GP) generation.

GP can be thought as generalizations of Guassian random vectors. They're fully characterized by mean and covariance functions and so, given these functions we can generate a GP using the (previously seen) method: factorization of the covariance matrix $\Sigma = AA^T$ and transformation of $\vec{Z} \sim N(\vec{0}, I)$ as $\vec{X} = \vec{\mu} + A\vec{Z}$.

3. Random Inputs Parametrization

Prior to any simulation of a given system, we have to properly characterize the random inputs. We learned how to represent random variables (computationally). In the case our inputs are random parameters it's straightforward to consider them as random variables and then to end in one of the previous cases. The problem is when the inputs are not finite dimensional, but infinite, i.e. random fields. We need a technique to reduce the dimension of an infinite dimensional object. The idea is to represent a random field $a(x;\omega)$ through its mean and covariance functions. More precisely, we can re-write the random field as a series constructed on its mean and variance (Karhunen-Loeve: $a(x;\omega) = \mathbb{E}[a](x) + \sum_{i=1}^{\infty} \sqrt{\lambda_i} b_i(x) Y_i(\omega)$) and then troncate the series whenever we want, obtaining N random objects that approximate the random field. We're using the spectral decomposition of an object. How can it work? Thanks to Mercer's theorem.

A (very) simple example of this procedure can be seen in the case of Gaussian random fields.

way to parametrize random infinite dimension objects

4. Monte Carlo Methods

Simulations can be used to study properties of the underlying statistical model: if we generate a large number of samples from a given model then the samples will reflect the statistical behaviour of the model. We focus on computing $\mathbb{E}[f(X)]$, where X is a random variable (or vector $\vec{X}$) and $f : \mathbb{R} \to \mathbb{R}$. We introduce the Monte Carlo method: given a random variable X we generate N iid replicas $X^{(1)}, .., X^{(N)}$ of X and we estimate $\mathbb{E}[X] := \mu$ (f identity) as $\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} X^{(i)}$ (assuming $Var(X) = \sigma^2 < \infty$). MC is simply based on LLN.

Monte Carlo to compute integrals.

We consider $Z = \psi(\vec{X}) = \psi(X_1, .., X_d)$, where $\vec{X} \sim f$ (joint distribution). We can approximate the integral $\mathbb{E}[Z] = \mu = \int_{\mathbb{R}^d} \psi(\vec{x}) f(\vec{x}) d\vec{x}$, trhough MC method, obtaining $\hat{\mu} := \frac{1}{N} \sum_{i=1}^{N} \psi(\vec{X}^{(i)})$. On the other hand, to approximate the integral $I = \int_{\mathbb{R}^d} \psi(\vec{x}) w(\vec{x}) d\vec{x}$, where $w : \mathbb{R} \to \mathbb{R}^+$, $\int_{\mathbb{R}^d} w = 1$ we can use the MC estimation $\hat{I} = \frac{1}{N} \sum_{i=1}^{N} \psi(\vec{X})^{(i)}$ with $\vec{X}^{(i)} \sim w$. So, we have a method for compute integrals with a rate of convergence of the order of $\frac{1}{\sqrt{N}}$, which is not so good, however the error is independent of the dimension d.

Smooth functions of expectations and delta method.

We can not only approximate $\mathbb{E}[Z]$, but also a general function of the expectation of a stochastic model $\vec{Z}$: $\zeta = f(\mathbb{E}[Z_1], .., \mathbb{E}[Z_m])$. The MC estimator of ζ is $\hat{\zeta} = f(\hat{\mu}_1, .., \hat{\mu}_m)$, where $\hat{\mu}_i = \frac{1}{N} \sum_{k=1}^{N} Z_i^k$ (k-th replica, i-th component). To estimate the error of $\hat{\zeta}$ we introduce the delta method (another version of the CLT based on the first-order Taylor expansion around $\vec{\mu} = \mathbb{E}[\vec{Z}]$). From here we can also construct a confidence interval.

5. Variance Reduction Techniques

We consider $Z = \psi(\vec{X}) = \psi(X_1, .., X_d)$, where $\vec{X} \sim f$ (joint distribution). We want to approximate $\mu = \mathbb{E}[Z]$. Trhough MC we obtain $\hat{\mu} := \frac{1}{N} \sum_{i=1}^{N} \psi(\vec{X}^{(i)}) = \frac{1}{N} \sum_{i=1}^{N} Z^{(i)}$, with the error: $|\mu - \hat{\mu}_{CMC}| \leq z_{1-\frac{\alpha}{2}} \frac{\sqrt{Var(Z)}}{\sqrt{N}}$. The only reducible part is $\sqrt{Var(Z)}$. To reduce it, instead of apply the mean estimator $\hat{\mu} = \hat{\mu}(Z)$ to Z, we apply it to a modified version $\tilde{Z}$ ($\mathbb{E}[\tilde{Z}] = \mathbb{E}[Z] = \mu$, $Var(\tilde{Z}) < Var(Z)$). How can we build the modified $\tilde{Z}$?

- **Antithetic Variables**

  The idea of this method is to generate $\frac{N}{2}$ pairs $(Z^{(i)}, Z^{(i+1)})$ of negatively correlated random variables instead of generating N independent random variables. In this way we generate $\hat{\mu}_{AV}$ such that $Var(\hat{\mu}_{AV}) = \frac{Var(Z) + Cov(Z^{(1)}, Z^{(2)})}{N} < \hat{\mu}_{CMC}$. However, a strong limitation of the method is the fact that to obtain a sample in which pairs of random variables are negatively correlated we have to assume that $\psi(\cdot)$ is non-increasing or non-decreasing.

- **Importance Sampling**

  The idea of this method is to transform $\mathbb{E}_f[Z] = \mathbb{E}_f[\psi(\vec{X})]$ in $\mathbb{E}_g[\frac{\psi(\vec{X}) f(\vec{X})}{g(\vec{X})}]$ and then procede with MC method, obtaining $\hat{\mu}_{IS}$. The efficiency of the method is based on $g(\cdot)$. More precisely: if we have to evaluate $\mathbb{E}[\psi(\vec{X})]$ we usually use $f(\cdot)$ (i.e. $\mathbb{E}_f[\psi(\vec{X})]$), but if $\psi(\cdot)$ has high values where $f(\cdot)$ puts low weight then we have a problem. The solution is to introduce the function $g(\cdot)$ which role is to fix these zones with high gap between $\psi(\cdot)$ and $f(\cdot)$.

- **Control Variates**

  The idea of this method is to apply the crude MC method, instead of on Z, on $\tilde{Z}_\alpha = Z + \alpha(Y - \mathbb{E}[Y])$, where Y is such that $\mathbb{E}[Y]$ is known and Y is strongly correlated with the variable Z. Under α_{opt} we have that $Var(\tilde{Z}_{\alpha_{opt}}) \leq Var(Z)$. Also, we can extend it to the case of multiple control variates $Y_1, .., Y_p$.

- **Stratification (or stratified sampling)**
  Another way to evaluate $\mathbb{E}[Z]$ over a domain Ω is to partition the domain as $\{\Omega_j\}_j$ (Ω_j are called strata) and sample on each part separately. Once we obtain $\{\hat{\mu}_j\}_j$ through the standard MC method (in every strata Ω_j we use MC method to determine $\hat{\mu}_j$), we compute $\hat{\mu}_{str}$ as a weighted sum of $\{\hat{\mu}_j\}_j$. The weightening is based on the probability of ending in Ω_j. How to choose how many strata? We can procede with a proportional allocation or with an optimal allocation.
- **Latin Hypercube Sampling**
  We want to evaluate $\mathbb{E}[Z]$, where $Z = \psi(X_1, .., X_d)$. According to stratification, we would like to stratify each variable X_j in s strata. However, this would lead to s^d strata, which is unaffordable for larger d. A way to overcome this problem is the Latin Hypercube Sampling. The idea of the LHS is to stratify each component X_j but not the whole sampling domain. N correlated points $\vec{X}^{(i)}$ are drawn in such way that each components is stratified within N strata and one point per stratum.

6. Quasi-Monte Carlo Methods

We introduce a class of sampling techniques that have more deterministic components. We go from the pure random sampling (pure Monte Carlo) to the low-discrepancy sampling (quasi-Monte Carlo). The goal is to compute $\mathbb{E}[Z] := \mu$, where $Z = \psi(X_1, .., X_d)$ and $\vec{X} = (X_1, .., X_d) \sim \mathcal{U}([0,1]^d)$, hence, a high-dimensional integration problem. The idea is to keep the sample average estimator structure but to turn to a pure deterministic sampling. How can we generate good point sets? Quasi-MC method chooses points as elements of a low-discrepancy sequence (a sequence has low-discrepancy if we can use it to obtain a good approximation of a given volume). There are severals measures of discrepancy. Thanks to the Koksma-Hlawka inequality we can bound the error of the quasi-MC method by the product of two terms: one depends on the function being integrated and the other is the discrepancy of the point set.

Part II: Forward Uncertainty Quantification & Sensitivity Analysis

1. Sensitivity Analysis

*Sensitivity analysis quantifies the effects of input parameters variation on outputs of interest, providing a criterium to rank the most influential input parameters. A **local approach** is based on the evaluation of the partial derivatives of the output w.r.t. each input parameter. Evaluating partial derivatives of complex outputs is not straightforward, and since the method is based on the evaluation of many points in the parameter space, this approach is not a good idea. However, if at each point we analyze the incremental ratio instead of the derivatives, we end up with a "good" method (screening method, one-at-the-time parameter variation).*
*A **global approach** (variance-based sensitivity analysis) relies instead on sampling over the whole parameter space. The idea is to introduce sensitivity indicators that explain the variance of the output in terms of:*

- ***first-order effects**: contribution of each factor singularly (separately). This index doesn't capture enough informations if the model is nonlinear in the inputs or if the model is non additive ($\exists$ iteractions)*
- ***total effects**: contribution of each factor complexively, including all the variance caused by iteractions (of any order) with any other input variables*

We can evalate these indeces considering the HDMR (High-Dimensional Model Representation), a useful expansion used to separate the different effects of the inputs on the output, and computing the needed integrals via quasi Monte Carlo method (crude MC is inefficient in this setting).

Part III : Inverse Uncertainty quantification

Safe toolbox for sensitivity analysis

Safetoolbox.info